

A decorative graphic in the top-left corner consisting of a grid of small squares in red, orange, and yellow, arranged in a pattern that tapers to the right.

# Postgre-DBA

**• REC** Проверить, идет ли запись

# Меня хорошо видно && слышно?



Тема вебинара

# Работа с большим объемом реальных данных



**Алексей Железной**

**Senior Data Engineer**

- Руководитель курсов DWH Analyst, ClickHouse для инженеров и архитекторов БД в OTUS
- Преподаватель курсов Data Engineer, DWH Analyst, PostgreSQL и пр. в OTUS

[LinkedIn](#)

# Правила вебинара



Активно  
участвуем



Задаем вопрос  
в чат или голосом



Вопросы вижу в чате,  
могу ответить не сразу

## Условные обозначения



Индивидуально



Время, необходимое  
на активность



Пишем в чат



Говорим голосом



Документ



Ответьте себе или  
задайте вопрос

# Маршрут вебинара

Обсудим термин “большие данные”

Изучим особенности работы с ними

Поработаем с ними в PostgreSQL

Рефлексия

# Цели вебинара

К концу занятия вы сможете

1. Понимать природу больших данных и знать места их обитания
2. Загружать большие данные в PostgreSQL
3. Иметь представление о том, как работать с большими данными в PostgreSQL

# Что такое большие данные?

# Big Data

## Большие данные - Big Data:

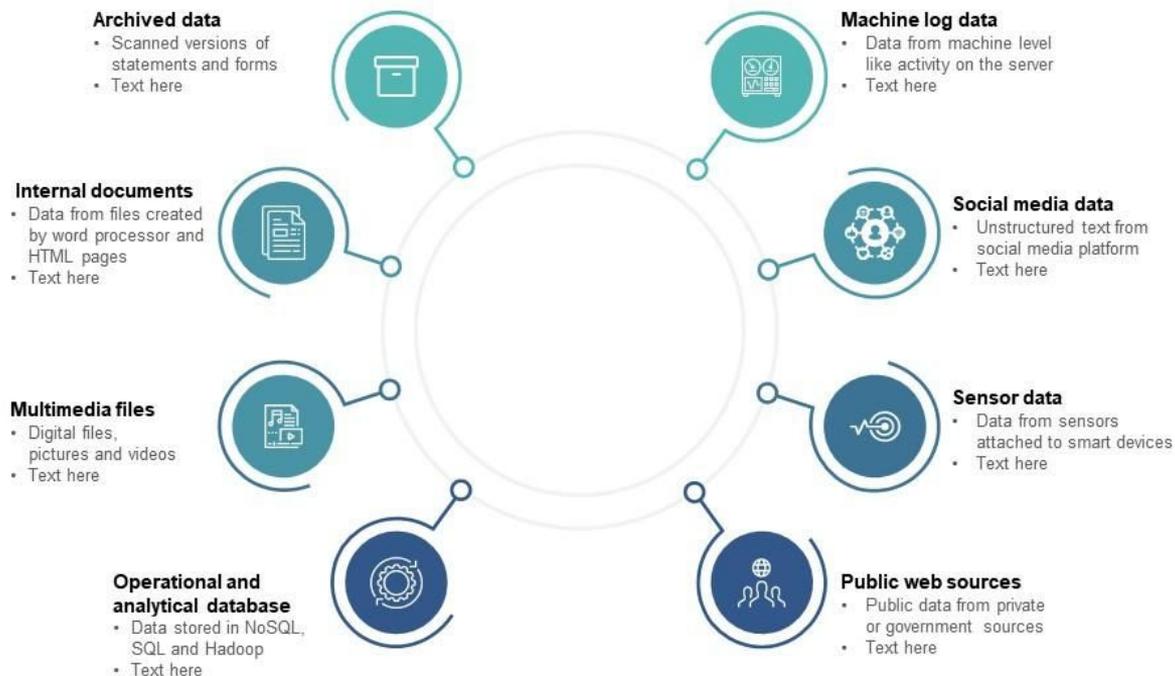
- серия подходов, инструментов и методов обработки структурированных и неструктурированных данных огромных объёмов и значительного многообразия
  - для получения воспринимаемых человеком результатов, эффективных в условиях непрерывного прироста, распределения по многочисленным узлам вычислительной сети
    - сформировавшихся в конце 2000-х годов, альтернативных традиционным системам управления базами данных и решениям класса Business Intelligence

# Примеры больших данных

- логи поведения пользователей в интернете
- GPS-сигналы от автомобилей для транспортной компании
- информация о транзакциях всех клиентов банка
- информация о всех покупках в крупной ритейл сети
- информация о движении товара в маркетплейсе
- и т.д.

# Источники больших данных

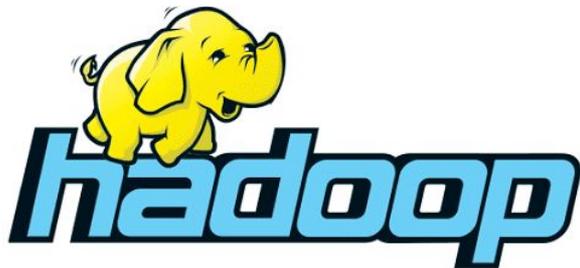
## Major Data Sources for Bigdata Strategy



# Принципы работы с большими данными

- Горизонтальная масштабируемость
  - большое количество узлов
- Отказоустойчивость
  - постоянное добавление и удаление узлов
- Локальность данных
  - данные распределены по множеству узлов
  - шардирование
  - дублирование

# Примеры систем для работы с Big Data



Google  
BigQuery

VERTICA



snowflake®



ClickHouse



Greenplum

# Вопросы?



Ставим "+",  
если вопросы есть



Ставим "-",  
если вопросов нет

# OLAP - просто о сложном

# OLAP

**OLAP = OnLine Analytical Processing** = аналитическая обработка данных в реальном времени. *Многомерные БД*

**O L A P**

**Processing** = Обрабатываются некие исходные данные...

**Analytical** = ... чтобы получить какие-то аналитические отчеты или новые знания...



**OnLine** = ... в реальном времени, практически без задержек на обработку.

# Бизнес-смысл OLAP

**Залог успешного бизнеса с точки зрения Big Data:**

- + Много данных (как фактовых, так и исторических)
  - + Проработанные механизмы сбора и обработки данных
  - + Крутые, мощные системы для их хранения, анализа
  - + Наглядные BI-системы для отображения информации
- 
- Принятие подготовленных, “правильных” бизнес-решений

*Сферы применения - в анализе тенденций, финансовой отчетности, прогнозировании продаж, бюджетировании и других целях планирования*

# Тех-смысл OLAP

- используется для построения отчетов на основе больших объемов накопленных исторических данных за огромные промежутки времени, но эти отчеты обновляются не слишком часто (\*)
- чаще всего это столбцовые СУБД (или поддерживающие column-orientation)
- выбирает данные быстро
- в центре находится таблица фактов, в которой находятся все показатели (сумма, кол-во) и ссылки на справочники (\*)
- чем больше столбцов, тем ниже скорость выполнения операций над строками (таких как добавление или изменение данных)
- больше про денормализацию

# OLTP - несложно о простом

# OLTP

**OLTP = OnLine Transactional Processing** - обработка транзакций в реальном времени.

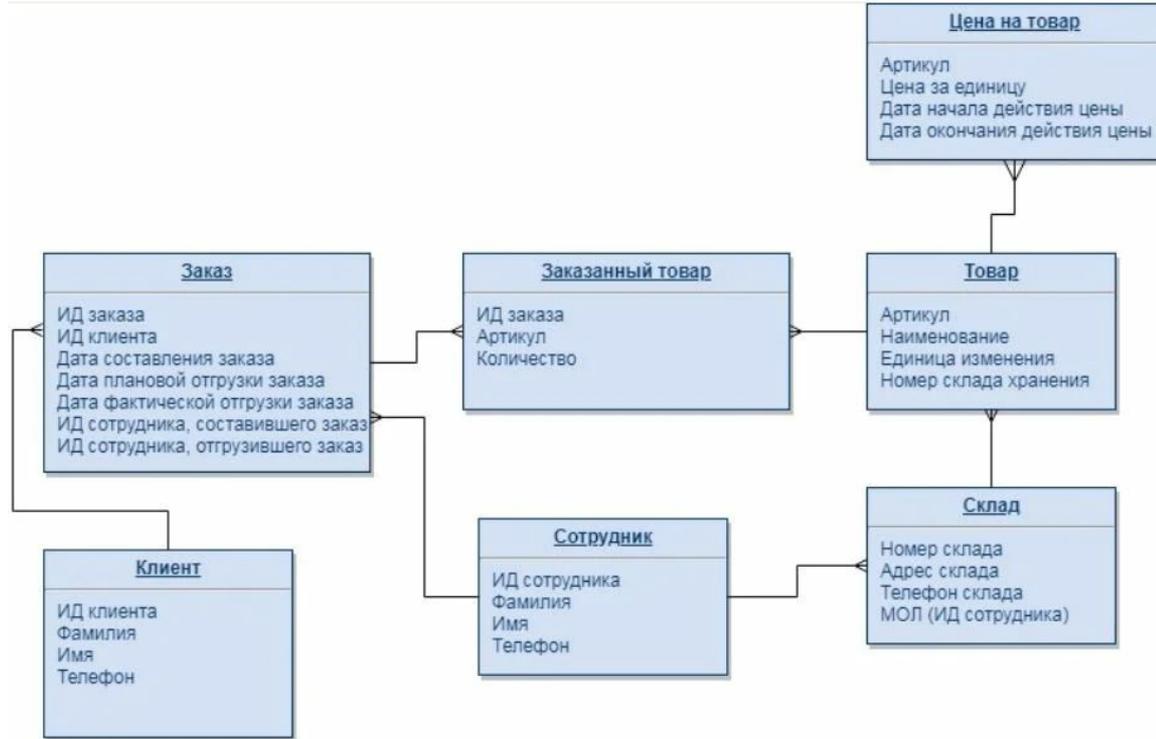
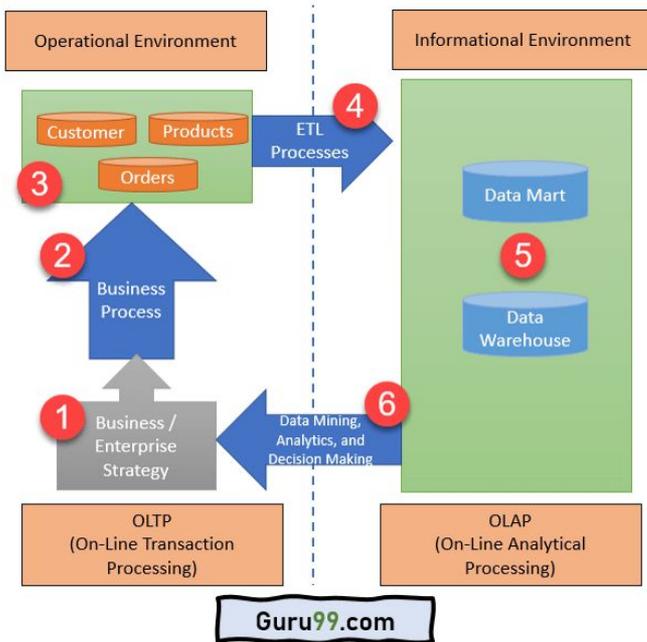
## *Реляционные БД*

- Одновременное выполнение нескольких транзакций (экономических, финансовых, цифрового взаимодействия), таких как интернет-банкинг, покупки, ввод заказов или отправка текстовых сообщений
- Задача - ввод, редактирование, удаление данных в режиме онлайн и их хранение
- Больше про нормализацию

# Особенности OLTP

- Нормализованные данные;
- Высокая интенсивность добавления и изменения данных;
- Большое количество одновременно активных пользователей (\*)
- Внесение данных и расчеты осуществляют пользователи системы;
- Содержат актуальные данные (\*)
- OLTP использует транзакции, которые включают небольшие объемы данных.
- Индексированные данные в базе данных могут быть легко доступны
- Трехуровневая архитектура, которая обычно состоит из уровня представления, уровня бизнес-логики и уровня хранилища данных

# Примеры OLTP



# Вопросы?



Ставим "+",  
если вопросы есть



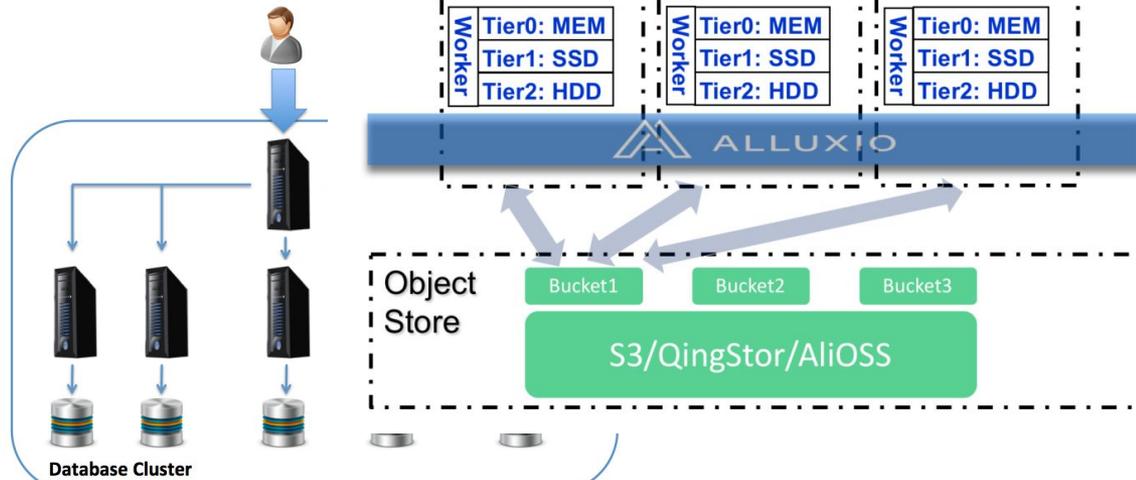
Ставим "-",  
если вопросов нет

# Теория аналитических СУБД

# Что такое MPP, и как это работает?

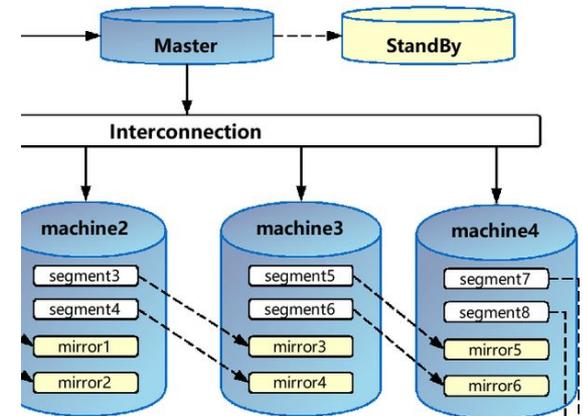
- Massive Parallelism
- Выделение ресурсов на узлах

MPP System Architecture



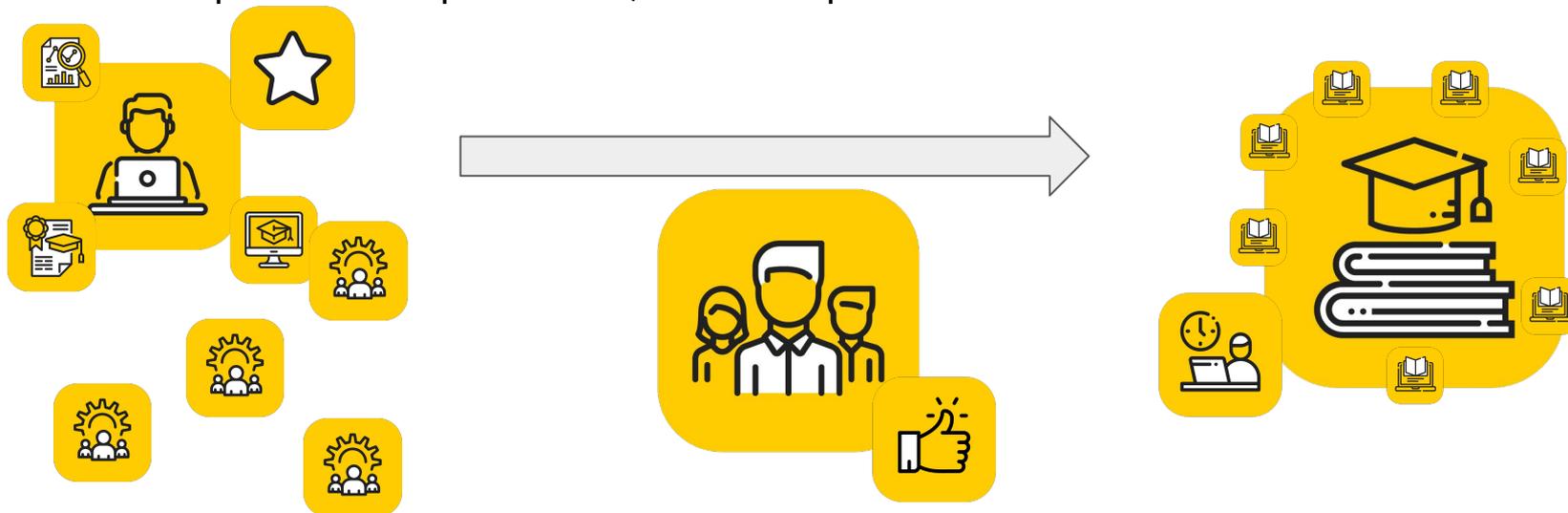
ных)

скольким различным



# Анализ больших данных: человеческий пример

Масштабировать по горизонтали, а не по вертикали



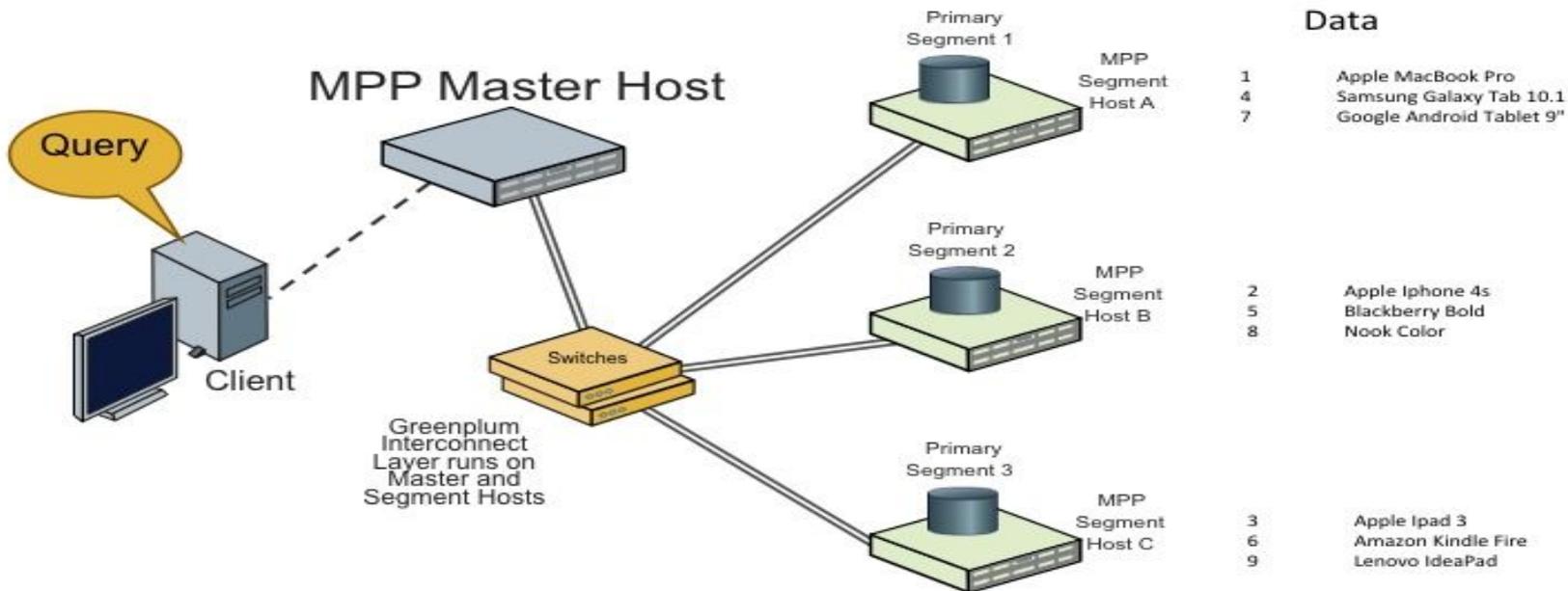
***Это массивно-параллельная обработка в действии, только с людьми, а не с компьютерами. Разделение простых, но больших задач на несколько сегментов и одновременная обработка этих сегментов будет намного быстрее, чем один человек, работающий в одиночку, независимо от того, насколько он квалифицирован.***

# Что такое MPP, и как это работает?

- **База данных MPP** — это тип базы данных или хранилища данных, в котором данные и вычислительная мощность распределяются между несколькими различными узлами (серверами) с одним ведущим узлом и одним или несколькими вычислительными узлами
- В MPP лидера (вас) называют ведущим узлом, работников библиотеки - вычислительными узлами
- Базы данных MPP можно масштабировать горизонтально, добавляя дополнительные вычислительные ресурсы (узлы)
- **Плюсы** — относительная быстрота обработки больших объемов данных (*Shared Nothing*), простота горизонтального масштабирования до сотен узлов, отказоустойчивость за счет зеркалирования и резервирования.
- **Минусы** — высокие требования к ресурсам, низкая производительность при большом объеме простых запросов, неоптимальное распределение сегментов

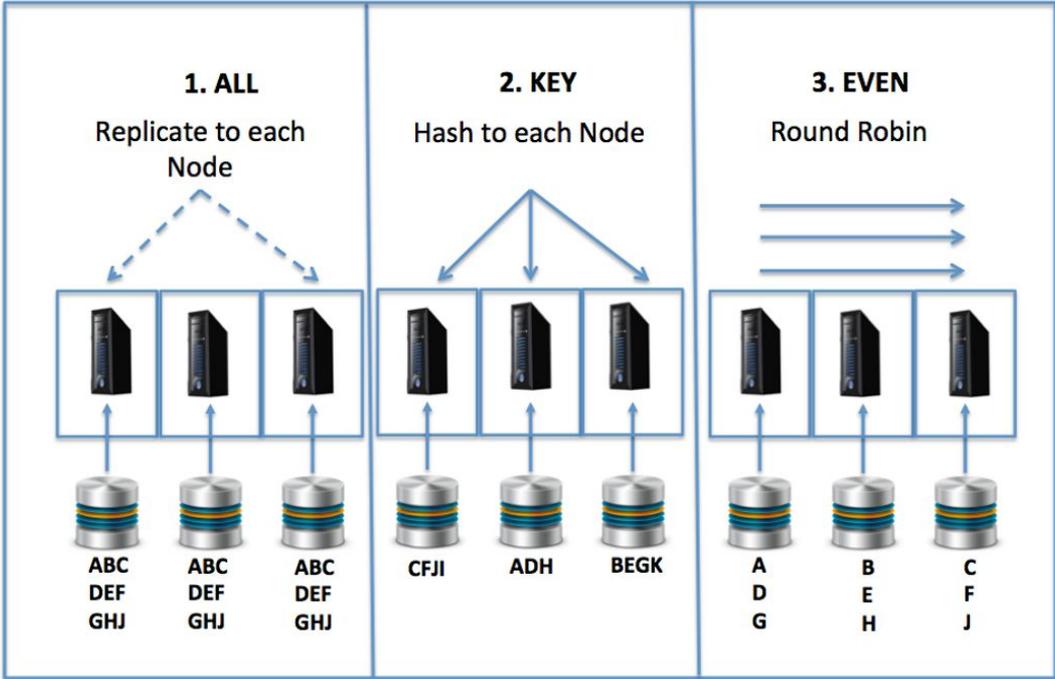
# Что такое MPP, и как это работает?

## Data Distribution



# MPP Data Distribution

## Three MPP Data Distribution Styles



# Представители MPP СУБД



Google BigQuery



# Вопросы?



Ставим “+”,  
если вопросы есть

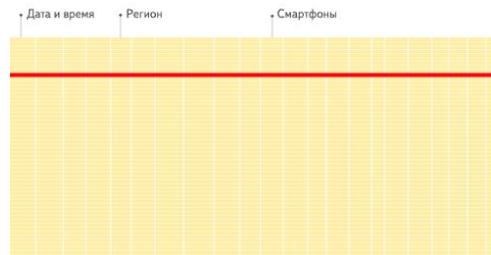
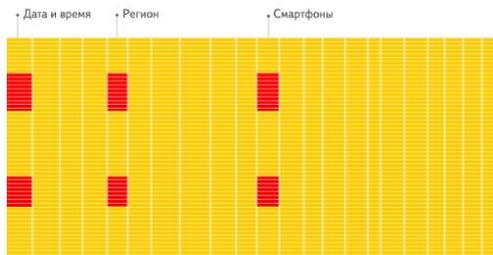


Ставим “-”,  
если вопросов нет

# Примеры систем для работы с большими данными

# ClickHouse в двух словах

- колоночная аналитическая БД
- написана на C++ в Яндексе
- open source (Apache License 2.0)
- НЕ ТОРМОЗИТ!



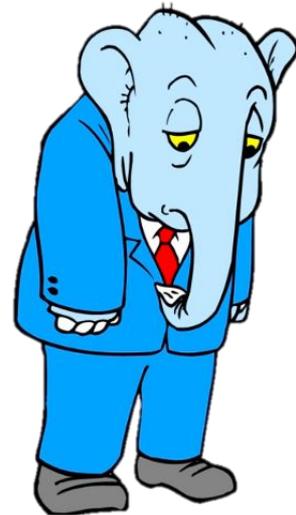
# Что ClickHouse предлагает из коробки

- переваривает огромные объемы информации
- векторная обработка запросов
- эффективная компрессия данных
- мощный SQL диалект
- репликация и шардирование из коробки
- множество типов данных (json, IP, UUID, Map, Array, etc)
- работа с геоданными, машинным обучением
- поддержка RabbitMQ, Kafka, MySQL, Postgres
- etc

# А что по PostgreSQL?

- PostgreSQL не предназначен для работы с большими данными, однако является прекрасным источником для них, как впрочем любая другая транзакционная СУБД

А почему нет (давайте подумаем)?



# Почему нет?

- транзакции и WAL
  - механизм транзакций мешает работе с большими данными
  - механизм WAL мешает работе с большими данными
- горизонтальное масштабирование
  - отсутствие эффективного механизма горизонтального масштабирования
  - отсутствие эффективного механизма шардирования данных

# Greenplum в двух словах

- **Greenplum** – open-source продукт, массивно-параллельная реляционная СУБД для хранилищ данных с гибкой горизонтальной масштабируемостью и столбцовым хранением данных на основе PostgreSQL.

*Greenplum нужна тем, у кого данных больше, чем очень много, то есть для работы с большими данными.*

*Greenplum нужна тем, кто не только хранит огромные объемы информации, но и активно с ними работает.*

# Архитектура Greenplum

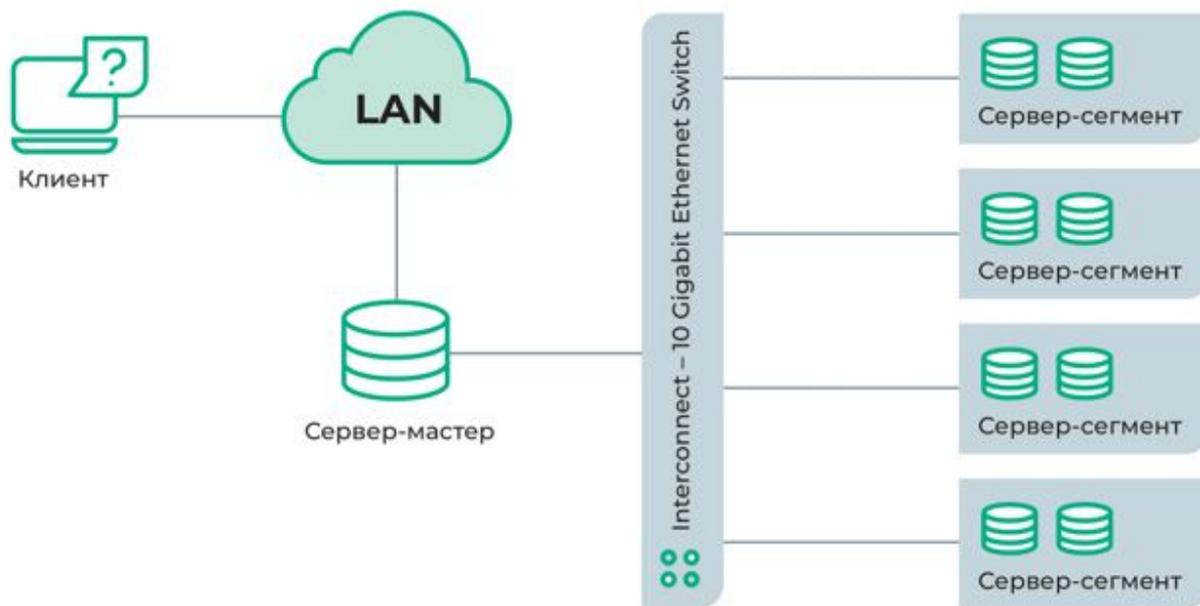


Рис. 3

# Вопросы?



Ставим "+",  
если вопросы есть



Ставим "-",  
если вопросов нет

# Загрузка больших данных в PostgreSQL

# Варианты загрузки

- Foreign Data Wrappers ([FDW](#))
- Foreign Data Wrappers in PostgreSQL and a closer look at [postgres\\_fdw](#)
- SQL copy
- [pgloader](#)
- [pg\\_bulkload](#)

# Postgres FDW

*As the name indicates, this feature allows a PostgreSQL database to treat tables in a remote PostgreSQL database as locally available tables.*

- История FDW началась с момента появления SQL/MED в составе спецификации стандарта ANSI SQL в 2003 году. Аббревиатура MED расшифровывается как "Управление внешними данными".
- По определению, "внешние данные" - это данные, к которым СУБД имеет доступ, но не управляет ими. Эта спецификация состоит из двух частей:
  - Foreign Table: речь идет о том, как получить доступ к внешним источникам данных и представить их в виде реляционных таблиц.
  - Datalink: расширяет функциональность систем баз данных, включая управление внешними файлами без необходимости хранения их содержимого непосредственно в базе данных

# Postgres FDW

- **Foreign Data Wrappers** - более новое явление, появившееся в PG 9+. В настоящее время в Postgres существует множество обёрток для внешних данных, которые работают с различными типами источников:
  - NoSQL-базами данных, платформами типа Twitter, форматами геопространственных данных и т.д.
- Postgres по умолчанию не поддерживает межбазовые запросы, даже на одном сервере.
- **postgres\_fdw** - это более или менее эквивалент dblink для доступа между серверами Postgres, с тем основным отличием, что postgres\_fdw соответствует стандартам SQL.

# Postgres FDW

- В настоящее время существует много готовых FDW, в том числе для Oracle, MySQL, Redis, MongoDB, ClickHouse, Kafka, Cassandra и RocksDB.
- Если нужный FDW еще не написан, вы можете [реализовать его самостоятельно](#).
- Foreign-Data Wrappers обеспечивают доступ к разным СУБД через одну точку доступа. Это удобно при выполнении аналитических запросов, а также с некоторой долей успеха позволяет горизонтально масштабировать PostgreSQL на запись.
- Еще FDW полезен при миграции на PostgreSQL.

## Разница dblink и FDW

# Перед загрузкой

- Отключить автокоммит
- Убрать индексы
- Убрать внешние ключи
- Увеличить параметр `maintenance_work_mem` (Задаёт максимальный объём памяти для операций обслуживания БД, в частности `VACUUM`, `CREATE INDEX` и `ALTER TABLE ADD FOREIGN KEY`)
- Увеличить `max_wal_size` (Максимальный размер, до которого может вырастать WAL во время автоматических контрольных точек)

# Работа с аналитическими запросами в PostgreSQL

- Партиционирование

```
CREATE TABLE measurement (  
    city_id          int not null,  
    logdate          date not null,  
    peaktemp        int,  
    unitsales       int  
    ) PARTITION BY RANGE (logdate);
```

- Витрины данных
- Модель данных
- Правильное индексирование

# Вопросы?



Ставим "+",  
если вопросы есть



Ставим "-",  
если вопросов нет

# Рефлексия

# Цели вебинара

К концу занятия вы сможете

1. Понимать природу больших данных и знать места их обитания
2. Загружать большие данные в PostgreSQL
3. Иметь представление о том, как работать с большими данными в PostgreSQL

# Список материалов для изучения

1. [Обзор гибких методологий проектирования DWH](#)

# Рефлексия



С какими впечатлениями уходите с вебинара?



Как будете применять на практике то, что узнали на вебинаре?

**Заполните, пожалуйста,  
опрос о занятии  
по ссылке в чате**

Спасибо за внимание!

# Приходите на следующие вебинары



**Алексей Железной**

**Senior Data Engineer**

- Руководитель курсов DWH Analyst, ClickHouse для инженеров и архитекторов БД в OTUS
- Преподаватель курсов Data Engineer, DWH Analyst, PostgreSQL и пр. в OTUS

[LinkedIn](#)